Condition Variables Solutions

Copyright © James Raynard 2020

Producer-Consumer Model

- Write a simple program that uses a condition variable to demonstrate the producer-consumer model

Copyright © James Raynard 2020

Condition Variable Scenario

- Typically, the consumer thread uses a unique_lock instance to lock the mutex, but the producer thread uses a lock_guard
- Why do the two threads use different lock types?
  - The consumer thread needs to be able to unlock the mutex. This is done inside the wait() call. In some applications, the waiting thread may need to directly unlock the mutex as well.
  - Therefore the mutex must support unlock()
  - Hence unique_lock is used

Copyright © James Raynard 2020

Condition Variable Scenario

- The producer thread only needs the mutex to protect the critical section. The extra flexibility and overhead of unique_lock is not needed here, so we use the simpler lock_guard
- We put the lock_guard and the critical section in their own scope
- The mutex will be automatically unlocked by the lock_guard destructor

Copyright © James Raynard 2020

Condition Variable Example

- Now reverse the order of the threads, so that the producer thread is started first. Add a sleep (say, half a second) before starting the consumer thread. What happens?
  - The producer thread completes before the reading thread runs
  - The consumer thread has not called wait() on the condition variable
  - The condition variable does not have any waiting threads to notify
  - The notification is "lost"
  - By the time the consumer thread calls wait(), the producer thread has completed
  - The condition variable does not receive any more notifications
  - The reading thread blocks for ever

Copyright © James Raynard 2020

Condition Variable with Predicate

- Modify your program to use a predicate in the call to wait()
- What happens?
  - The notification is now received by the consumer thread

Copyright © James Raynard 2020